國 立 交 通 大 學

應用數學系

碩　士　論　文

Poisson 方程在不規則區域的高階緊緻差分法

Higher-order compact difference scheme for the Poisson equation on irregular domains

研 究 生：陳莉君

指導老師：賴明治　教授

中 華 民 國 九 十 四 年 六 月

Poisson 方程在不規則區域的高階緊緻差分法

Higher-order compact difference scheme for the Poisson equation on irregular domains

研 究 生：陳莉君　　Student：Li-Jun Chen

指導教授：賴明治　　Advisor：Ming-Chih Lai

國 立 交 通 大 學
應 用 數 學 系
碩 士 論 文

A Thesis
Submitted to Department of Applied Mathematics
College of Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in
Applied Mathematics
June 2005

Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 四 年 六 月

Poisson 方程在不規則區域的高階緊緻差分法

學生：陳莉君　　　　　　　　　　　　　　　　　　指導教授：賴明治

國立交通大學　應用數學　學系（研究所）碩士班

摘　　　要

在這篇論文中，我們考慮在不規則區域上有著 Dirichlet 邊界條件的二維 Poisson 方程，而此不規則區域將被限制在一個矩形區域中來做計算。在原不規則區域內的網格點，我們的高階精度方法是用標準的緊緻九點格式來離散此 Poisson 方程，但對於靠近邊界的點將要做些特別處理。在這些需要另外做處理的點，我們利用外插法來造出仿真的值。使用的外插法有常數、線性以及二次的外插法，並將分別可以得到一階、二階以及三階的精度。其中常數及線性的外插法，可以使要解的線性系統仍保持對稱性。

Higher-order compact difference scheme for the Poisson equation on irregular domains

student：Li-Jun Chen Advisors：Dr. Ming-Chih Lai

Department（Institute）of Applied Mathematics
National Chiao Tung University

ABSTRACT

In this thesis, we consider the 2D Poisson equation subject to Dirichlet boundary conditions on an irregular domain. The region of interest is embedded in a rectangular domain. For our higher-order accurate scheme, at internal grid points, the Poisson equation is discretized with the standard compact nine point stencil with special treatment at the edges. At the irregular point, we define ghost value constructed by extrapolations. This yields first, second and third order accuracy in the case of the constant, linear and quadratic extrapolations, respectively. In the case of constant and linear extrapolations, the linear system is symmetric.

誌　　謝

這篇論文的完成必須感謝許多協助與支持我的人。首先，感謝我的指導教授 賴明治老師這兩年來的指導與勉勵，在學問與待人處事方面，都讓我受惠良多，謹此致上我最誠摯的敬意與謝意。口試期間，承蒙葉立明老師、黃聰明老師及吳金典老師費心審閱並提供許多寶貴之意見，使本論文之完稿得以更加齊備，永誌於心。

研究所求學過程中，感謝昱豪學長總在我遇到問題的時候，給予我意見以及耐心的指導；接著感謝同窗好友雅靜與研究所同學們的關心與協助，這些日子裡的互相幫忙、互相砥礪，深厚的情誼永難忘懷。此外，謝謝球隊的舉卿學姊、雅文學姊、曲敏、乃心以及學妹們，因為有你們碩士班兩年生涯能夠如此充實愉快的度過，與你們一起的點點滴滴，將是我永遠的回憶。

最後，要感謝我最敬愛的父母，感謝你們在我學習的路上給予我最大的支持以及自由的學習環境，讓我可以無後顧之憂的學習。再次感謝所有幫助過我、關心我的人，謝謝你們！

目　　錄

1 Introduction

In this thesis, we consider the solution of the Poisson equation on an irregular domain, subject to Dirichlet boundary conditions. The Poisson equation subject to Dirichlet boundary conditions on an irregular domain can be treated by embedding the region in a rectangular domain and solving by finite differences over the domain. The crucial issue is the discretization of the boundaries of the irregular domain.

There are many other approaches to this problem in the literature. In [4], the authors solved a variable coefficient Poisson equation in the presence of an irregular interface where Dirichlet boundary conditions were imposed. They used a finite volume method that results in a non-symmetric discretization matrix. Both multi-grid methods and adaptive mesh refinement were used. In [5], this non-symmetric discretization was coupled to a volume of fluid front tracking method in order to solve Stefan problem.

In [9], the basic idea of the ghost fluid method [7] was employed to develop a first-order-accurate symmetric finite difference scheme based on the Cartesian grid to solve a variable Poisson equation in the presence of an irregular interface. Subsequently in [1], the approach in [9] was modified to obtain a second order accurate symmetric finite difference scheme based on the Cartesian grid to solve a variable Poisson equation with a Dirichlet boundary condition. The modification used the signed distance level set function to obtain a linear interpolation from the boundary value and the solution values in coordinate-wise directions to determine the ghost fluid values.

The intention of this paper is to extend the idea of [2,11]. In [2], the authors exploit the methodology of [1] to derive a fourth order accurate finite difference discretization for the Laplace equation on irregular domains. But in order to guarantee a fourth order accurate, the difference scheme that the authors used was a standard long stencil. The primary objective of this paper is to keep higher-order accurate and to make the scheme to be a compact one.

The rest of the paper is organized as follows: In Section 2 we deal with the 1-D Poisson equation with Dirichlet boundary conditions and try to solve Neumann boundary problem as well. In Section 3 we extend the methodology discussed in Section 2 to two spatial dimensions. Numerical examples are presented in Section 4 before we conclude with a summary in Section 5.

2 One-dimensional Poisson equation

We consider a Cartesian computational domain, $\Omega = [a, b]$, with a lower dimensional interface, Γ, that divides the computational domain into disjoint pieces, Ω^- and Ω^+. The 1-D Poisson equation is given by

$$T_{xx} = f, x \in \Omega^- = [a, x_I]. \tag{2.1}$$

A uniform grid is taken over $[a, b]$. Dirichlet boundary conditions or Neumann boundary conditions are assumed given at two boundary points $x = a$ and x_I, x_I typically is not grid point. In the other subdomain we set $T = 0$, so that we have

$$\begin{cases} T_{xx} = f & x \in \Omega^- \\ T = 0 & x \in \Omega^+ \end{cases} . \tag{2.2}$$

In general there is a discontinuity at x_I.

The solution to the Poisson equation is computed at the grid points and is written as $T_i = T(x_i)$. We consider the fourth order discretization :

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + O(\Delta x^4) = (T_{xx})_i + \frac{\Delta x^2}{12}(T_{xxxx})_i$$
$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} \approx \frac{f_{i+1} + 10f_i + f_{i-1}}{12} \tag{2.3}$$

For each unknown, T_i , Eq(2.3) is used to fill in one row of a matrix creating a linear system of equations. This discretization is valid if all the node values belong to the same domain, but needs to be modified otherwise. For example, suppose the interface location, x_I is located in between the nodes x_i and x_{i+1} (see Fig.1) and suppose that we seek to write the equation satisfied by T_i. Since the solution is not defined across the interface, we need valid values for T_{i+1} and f_{i+1} that *emulate* the behavior of the solution defined to the left of the interface. We achieve this by defining *ghost values* T^G_{i+1} and f^G_{i+1} constructed by extrapolating the value of T across the interface. The discretization for such points in the neighborhood of the interface is rewritten as

$$\frac{T^G_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} \approx \frac{f^G_{i+1} + 10f_i + f_{i-1}}{12} \tag{2.4}$$

In the remainder of this section, we describe how to construct the ghost values T^G_{i+1} and f^G_{i+1} more precisely.

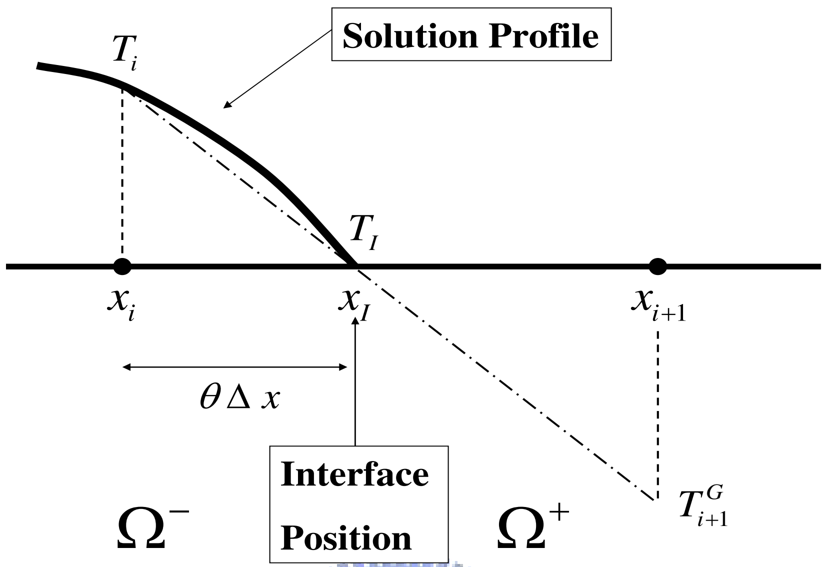

Figure 1: Definition of the ghost cells with linear extrapolation

2.1 Dirichlet boundary condition

In this section, we consider the situation that Dirichlet boundary conditions are given at two boundary points, $T(a) = T_a$ and $T(x_I) = T_I$. We first construct an interpolant $\widetilde{T}(x)$ of $T(x)$ on the left of the interface, such that $\widetilde{T}(0) = T_i$, and then we define $T_{i+1}^G = \widetilde{T}(\Delta x)$. Fig.1 illustrates the definition of the ghost cell in the case of the linear extrapolation.

We consider constant, linear, quadratic and cubic extrapolations defined by:

- **Constant extrapolation**
  Take $\widetilde{T}(x) = d$ with:

  1. $d = T_I$

- **Linear extrapolation**
  Take $\widetilde{T}(x) = cx + d$ with:

  1. $\widetilde{T}(0) = T_i$
  2. $\widetilde{T}(\theta\Delta x) = T_I$

- **Quadratic extrapolation**
  Take $\widetilde{T}(x) = bx^2 + cx + d$ with:

  1. $\widetilde{T}(-\Delta x) = T_{i-1}$
  2. $\widetilde{T}(0) = T_i$
  3. $\widetilde{T}(\theta\Delta x) = T_I$

- **Cubic extrapolation**
  Take $\widetilde{T}(x) = ax^3 + bx^2 + cx + d$ with:

  1. $\widetilde{T}(-\Delta x) = T_{i-1}$
  2. $\widetilde{T}(0) = T_i$
  3. $\widetilde{T}(\theta\Delta x) = T_I$
  4. $\widetilde{T}''(\theta\Delta x) = f_I$

Then we get

$$T_{i+1}^G = T_I \tag{2.5}$$

$$T_{i+1}^G = (1 - \frac{1}{\theta})T_i + \frac{1}{\theta}T_I \tag{2.6}$$

$$T_{i+1}^G = \frac{1-\theta}{1+\theta}T_{i-1} + \frac{2(\theta-1)}{\theta}T_i + \frac{2}{\theta(1+\theta)}T_I \tag{2.7}$$

and

$$T_{i+1}^G = \frac{(1-\theta)(2\theta-1)}{(2\theta+1)(\theta+1)}T_{i-1} + \frac{4(\theta-1)}{2\theta+1}T_i + \frac{6}{(2\theta+1)(\theta+1)}T_I + \frac{1-\theta}{2\theta+1}h^2 f_I \tag{2.8}$$

which are defined by constant, linear, quadratic and cubic extrapolations, respectively.

Similarly, we construct an interpolant $\widetilde{f}(x)$ of $f(x)$. The definitions of constant, linear and quadratic extrapolations are the same as $\widetilde{T}$. So, we have

$$f_{i+1}^G = f_I \tag{2.9}$$

$$f_{i+1}^G = (1 - \frac{1}{\theta})f_i + \frac{1}{\theta}f_I \tag{2.10}$$

and

$$f_{i+1}^G = \frac{1-\theta}{1+\theta}f_{i-1} + \frac{2(\theta-1)}{\theta}f_i + \frac{2}{\theta(1+\theta)}f_I \tag{2.11}$$

which are defined by constant, linear and quadratic extrapolations, respectively. But there is a little different to the cubic extrapolation :

- **Cubic extrapolation**
  Take $\widetilde{f}(x) = ax^3 + bx^2 + cx + d$ with:

  1. $\widetilde{f}(-2\Delta x) = f_{i-2}$
  2. $\widetilde{f}(-\Delta x) = f_{i-1}$
  3. $\widetilde{f}(0) = f_i$
  4. $\widetilde{f}(\theta\Delta x) = f_I$

then we have

$$f_{i+1}^G = \frac{\theta-1}{\theta+2} f_{i-2} + \frac{(3-3\theta)}{\theta+1} f_{i-1} + \frac{3\theta-3}{\theta} f_i + \frac{6}{\theta^3+3\theta^2+2\theta} f_I \tag{2.12}$$

which is defined by cubic extrapolation. In these equations $\theta = (x_I - x_i)/\Delta x$ refers to the cell fraction occupied by the subdomain Ω^-. This yields first, second, third and fourth order accuracy in the case of the constant, linear, quadratic and cubic extrapolations, respectively.

If we were solving for the domain Ω^+, the equation satisfied by T_{i+1} and f_{i+1} requires the definition of the ghost cells T_i^G and f_i^G. In this case, we write $T_i^G = \widetilde{T}(\Delta x)$ and $f_i^G = \widetilde{f}(\Delta x)$ with the definition for $\widetilde{T}$ modified as follows: $\theta = (x_{i+1} - x_I)/\Delta x$, T_i is replaced by T_{i+1}, f_i is replaced by f_{i+1}, T_{i-1} is replaced by T_{i+2} and f_{i-1} is replaced by f_{i+2}.

We note that the construction of $\widetilde{T}$ and $\widetilde{f}$ cannot be arbitrary. It is obviously limited by the number of points within the domain, but also by how close the interface from a grid node. The latter restriction comes from the fact that, as $\theta \to 0$, the behavior of the interpolant deteriorates.

In the case of the constant extrapolation, the corresponding matrix

$$A_{Constant} = \begin{array}{cl} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 0 \\ & & & 0 & 1 \end{bmatrix} & \begin{array}{l} \\ \\ \\ \to i\text{th row} \\ \to (i+1)\text{th row} \end{array} \end{array}$$

is symmetric and diagonally dominant. In the case of the linear extrapolation, the corresponding matrix

$$A_{Linear} = \begin{array}{cl} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -1-\frac{1}{\theta} & 0 \\ & & & 0 & 1 \end{bmatrix} & \begin{array}{l} \\ \\ \\ \to i\text{th row} \\ \to (i+1)\text{th row} \end{array} \end{array}$$

is symmetric and diagonally dominant. This allows for the use of fast iterative solvers such as preconditioned conjugate gradient. But in the case of the quadratic and cubic extrapolations, the corresponding matrices

$$A_{Quadratic} = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1+\frac{1-\theta}{1+\theta} & -2+\frac{2(\theta+1)}{\theta} & 0 \\ & & & 0 & 1 \end{bmatrix} \begin{array}{l} \\ \\ \\ \rightarrow i\text{th row} \\ \rightarrow (i+1)\text{th row} \end{array}$$

and

$$A_{Cubic} = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1+\frac{(1-\theta)(2\theta-1)}{(2\theta+1)(\theta+1)} & -2+\frac{4(\theta-1)}{2\theta+1} & 0 \\ & & & 0 & 1 \end{bmatrix} \begin{array}{l} \\ \\ \\ \rightarrow i\text{th row} \\ \rightarrow (i+1)\text{th row} \end{array}$$

are non-symmetric. So the non-symmetric linear system is solved with a **BiCGSTAB** (see e.g. [10]) using an incomplete LU factorization for the preconditioner. For the linear solver that require an initial guess, setting all T_i identically zero is usually sufficient.

2.2 Neumann boundary condition

We take little effort to replace the Dirichlet boundary conditions with Neumann boundary conditions, reformulating the Poisson equation as

$$\begin{cases} T_{xx} = f, & x \in [a, x_I] \\ T_x(a) = \alpha, T_x(x_I) = \beta \end{cases} . \tag{2.13}$$

So, we needs to do some modifications of the extrapolations $\widetilde{T}(x)$. We discuss linear, quadratic and cubic extrapolations defined by:

- **Linear extrapolation**
  Take $\widetilde{T}(x) = cx + d$ with:
  1. $\widetilde{T}(0) = T_i$
  2. $\widetilde{T}'(\theta\Delta x) = \beta$
- **Quadratic extrapolation**
  Take $\widetilde{T}(x) = bx^2 + cx + d$ with:

1. $\widetilde{T}(-\Delta x) = T_{i-1}$
2. $\widetilde{T}(0) = T_i$
3. $\widetilde{T}'(\theta\Delta x) = \beta$

- **Cubic extrapolation**
  Take $\widetilde{T}(x) = ax^3 + bx^2 + cx + d$ with:

  1. $\widetilde{T}(-\Delta x) = T_{i-1}$
  2. $\widetilde{T}(0) = T_i$
  3. $\widetilde{T}'(\theta\Delta x) = \beta$
  4. $\widetilde{T}''(\theta\Delta x) = f_I$

The discussion above leads naturally to

$$T_{i+1}^G = T_i + \beta\Delta x \tag{2.14}$$

$$T_{i+1}^G = \frac{1-2\theta}{1+2\theta}T_{i-1} + \frac{4\theta}{1+2\theta}T_i + \frac{2\beta\Delta x}{1+2\theta} \tag{2.15}$$

and

$$\begin{aligned} T_{i+1}^G = {} & \frac{-3\theta^2+3\theta-1}{3\theta^2+3\theta+1}T_{i-1} + \frac{6\theta^2+2}{3\theta^2+3\theta+1}T_i + \frac{6\theta}{3\theta^2+3\theta+1}\beta\Delta x \\ & + \frac{-3\theta^2+1}{3\theta^2+3\theta+1}f_I\Delta x^2 \end{aligned} \tag{2.16}$$

which are defined by linear, quadratic and cubic extrapolations, respectively. And the coefficient matrices of the linear system are

$$A_{Linear} = \begin{array}{c} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -1 & 0 \\ & & & 0 & 1 \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \\ \rightarrow i\text{th row} \\ \rightarrow (i+1)\text{th row} \end{array}$$

$$A_{Quadratic} = \begin{array}{c} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1+\frac{1-2\theta}{1+2\theta} & -2+\frac{4\theta}{1+2\theta} & 0 \\ & & & 0 & 1 \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \\ \rightarrow i\text{th row} \\ \rightarrow (i+1)\text{th row} \end{array}$$

and

$$A_{Cubic} = \begin{array}{c} \left[\begin{array}{ccccc} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1+\frac{-3\theta^2+3\theta-1}{3\theta^2+3\theta+1} & -2+\frac{6\theta^2+2)}{3\theta^2+3\theta+1} & 0 \\ & & & 0 & 1 \end{array}\right] \end{array} \begin{array}{l} \\ \\ \\ \rightarrow i\text{th row} \\ \rightarrow (i+1)\text{th row} \end{array}$$

which are obtained by linear, quadratic and cubic extrapolations, respectively.The extrapolations $\widetilde{f}$ of f don't have any modifications. Because we have the value of f on the boundary exactly.

This yields first, second and third order accuracy in the case of the linear, quadratic and cubic extrapolations, respectively. But this method has some shortcomings. It is unable to expand this method to solve two-dimensional Poisson equation with Neumann boundary conditions. Because we do not have enough information on the boundary to use this method.

The overall accuracy for T and the nature of the resulting linear system is determined by the degree of the interpolation function $\widetilde{T}$, which is summarized in Tab. 1.

Dirichlet boundary condition

Degree of extrapolation	Order of accuracy	Linear system
Constant	First	Symmetric
Linear	Second	Symmetric
Quadratic	Third	Non-symmetric
Cubic	Fourth	Non-symmetric

Neumann boundary condition

Degree of extrapolation	Order of accuracy	Linear system
Linear	First	Symmetric
Quadratic	Second	Non-symmetric
Cubic	Third	Non-symmetric

Table 1: Order of accuracy and nature of the linear system corresponding to the constant, linear, quadratic and cubic case

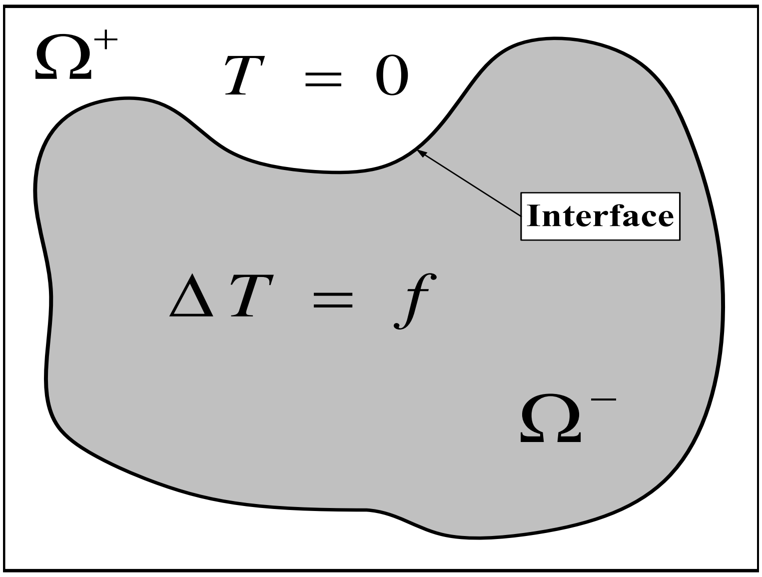

Figure 2: An irregular interface Γ dividing the domain Ω into two subdomain Ω^- and Ω^+.

3 Two-dimensional Poisson equation

Consider the two spatial dimension Poisson equation

$$\Delta T = f(x, y)$$

and let Ω^- be any irregular 2-D shape inscribed within a rectangle with boundary Γ at which Dirichlet conditions $T(x, y) = g(x, y)$ are specified.We can regard the boundary Γ as a interface that divies the domain Ω into two disjoint pieces, Ω^- and Ω^+(see Fig.2). As $T = 0$ outside the physical domain, there may be jumps on Γ.

We use the standard compact nine point stencil scheme, the 9-point Laplacian, denote by $\Delta_9 T_{i,j}$,

$$\Delta_9 T_{i,j} = \frac{1}{6h^2}[4T_{i-1,j}+4T_{i+1,j}+4T_{i,j-1}+4T_{i,j+1}+T_{i-1,j-1}+T_{i-1,j+1}+T_{i+1,j-1}+T_{i+1,j+1}-20T_{i,j}]$$

If we apply this to the true solution and expand in Taylor series we find that

$$\Delta_9 T_{i,j} = \Delta T_{i,j} + \frac{h^2}{12}[(T_{xxxx})_{i,j} + 2(T_{xxyy})_{i,j} + (T_{yyyy})_{i,j}] + O(h^4).$$

The additional terms lead to a very nice form for the dominant error term, since

$$T_{xxxx} + 2T_{xxyy} + T_{yyyy} = \Delta(\Delta T).$$

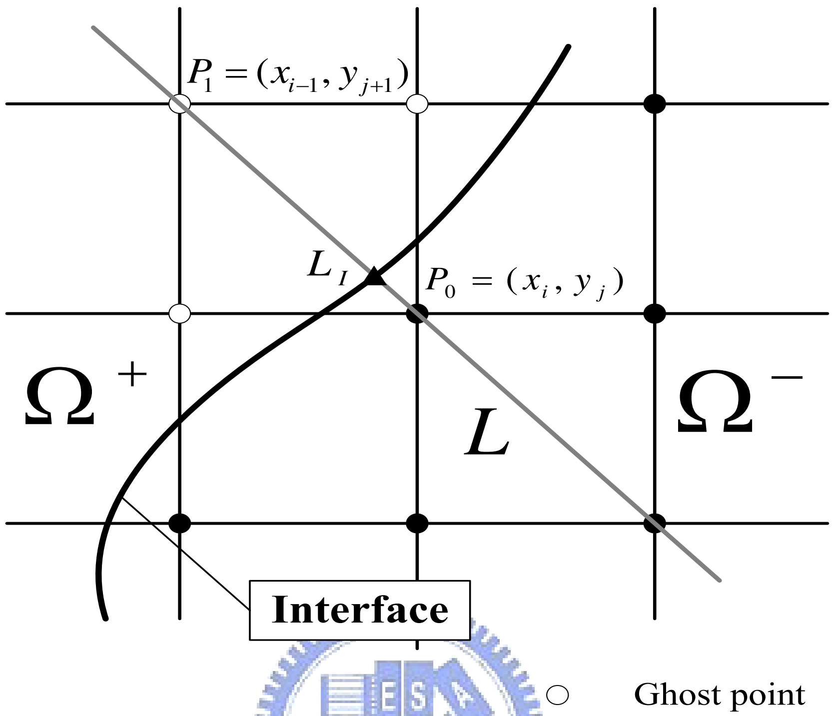

Figure 3: A diagram of an irregular grid point P_0.

This is the Laplacian of Laplacian of T which is known as the *biharmonic*. Because we are solving $\Delta T = f$, then we have

$$T_{xxxx} + 2T_{xxyy} + T_{yyyy} = \Delta f.$$

Hence we can compute the dominant term in the truncation error easily from the known function f without knowing the true solution T to the problem. So,we can obtain a fourth-order accurate method of the form

$$\Delta_9 T_{i,j} \approx \frac{f_{i-1,j} + f_{i+1,j} + f_{i,j-1} + f_{i,j+1} + 8f_{i,j}}{12}. \tag{3.17}$$

This discretization is valid if all the node values belong to the same domain, but needs to be modified otherwise. The methodology discussed in Section 2.1 extends naturally to two spatial dimensions. Before we modify the Eq.(3.17), we define a grid node (x_i, y_j) as *regular* if all neighbouring nodes are on the same side of the interface.On contrary, a grid node (x_i, y_j) as *irregular* if at least one adjacent node is on the other side of the interface. In order to illustrate our methodology, we suppose

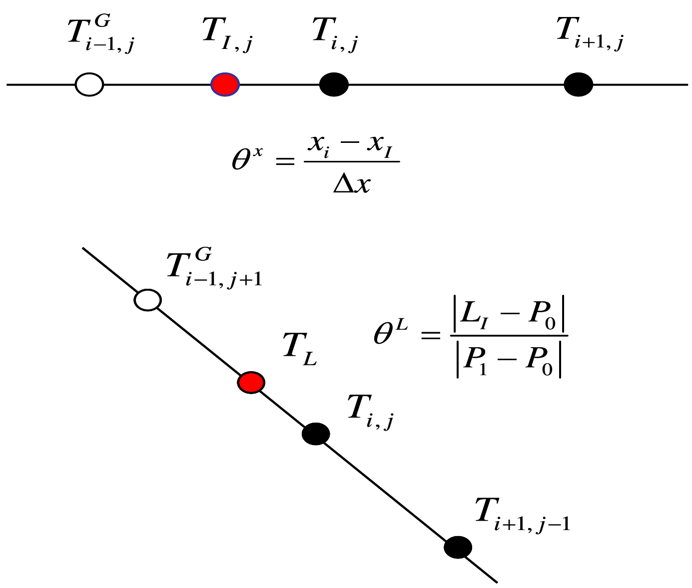

Figure 4: The ghost values $T^G_{i-1,j}$ and $T^G_{i-1,j+1}$ which were constructed along x-direction and the line segment L, respectively.

$P_0 = (x_i, y_j)$ is a irregular grid node and the distribution of its neighbouring nodes were displayed in Fig.3.

The discretization for the irregular point $P_0 = (x_i, y_j)$ is then written as

$$\frac{1}{6h^2}[4T^G_{i-1,j}+4T_{i+1,j}+4T_{i,j-1}+4T^G_{i,j+1}+T_{i-1,j-1}+T^G_{i-1,j+1}+T_{i+1,j-1}+T_{i+1,j+1}-20T_{i,j}]$$

$$\approx \frac{f^G_{i-1,j}+f_{i+1,j}+f_{i,j-1}+f^G_{i,j+1}+8f_{i,j}}{12}.$$

About the ghost values $T^G_{i-1,j}$ and $f^G_{i-1,j}$,we consider the left arm of the stencil, i.e. the line segment connecting (x_{i-1}, y_j) and (x_i, y_j). We first find the interface location (x_I, y_j) that is the intersection point of the interface and the line segment. In order to find (x_I, y_j), we solve a nonlinear equation. In our example section we use Newton's Method to solve nonlinear equation. Then we define $\theta^x = \frac{x_i - x_I}{\Delta x}$. So we can construct a constant, linear, or quadratic extrapolation $\widetilde{T}^x$ and $\widetilde{f}^x$ of T and f in the x-direction. The procedure to find $T^G_{i,j+1}$ and $f^G_{i,j+1}$ are similar.

The remainder ghost value $T^G_{i-1,j+1}$, we consider the line segment, L, connecting (x_{i-1}, y_{j+1}) and (x_i, y_j). Because the line segment and the interface are crossing in a point, so we have the interface locate L_I(see Fig.3). Then we define $\theta^L = \frac{|L_I - P_0|}{|P - P_1|}$. Finally, we can construct a constant, linear, or quadratic extrapolation $\widetilde{T}^L$ of T

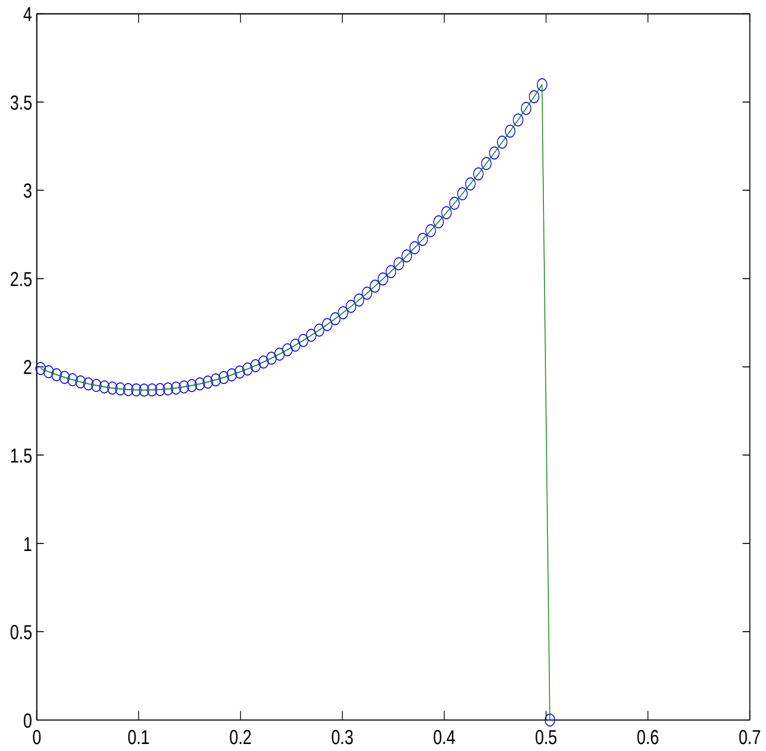

Figure 5: 1D Poisson equation, $T_{xx} = f$,on [0,0.5] with Dirichlet boundary conditions. The exact solution is $T = x^7 - x^3 + 12x^2 - 2.5x + 2$. The grid size is 64 and the ghost cells are defined by quadratic extrapolation.

along the line segment L(see Fig.4). Note that on irregular domains, the number of available grid nodes within the domain might limit the extrapolation to a lower degree for some grid resolution.

4 Examples

We test our methodology on the following examples. In the case where the linear system is symmetric, we use a preconditioned conjugate gradient method with an incomplete Cholesky preconditioner. In the case where the linear system is non-symmetric, we use the **BICGSTAB** method. The order of the scheme is given as

$$\text{order} = \left| \frac{\log(\|Error_n\|_\infty / \|Error_{2n}\|_\infty)}{\log(2)} \right| .$$

4.1 Example 1

Consider $T_{xx} = f$ on $\Omega = [0, 0.5]$ with an exact solution of $T = x^7 - x^3 + 12x^2 - 2.5x + 2$. The computational domain is discretized into cell of size Δx where the

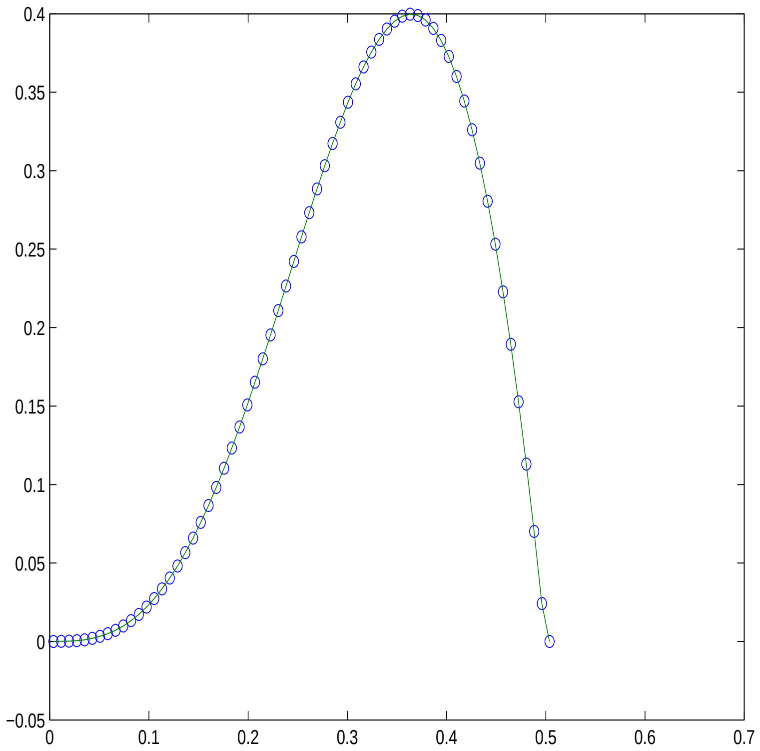

Figure 6: 1D Poisson equation, $T_{xx} = f$,on [0,0.5] with Neumann boundary conditions. The exact solution is $T = 4x^2 \sin(2\pi x)$. The grid size is 64 and the ghost cells are defined by cubic extrapolation.

cell centers are referred to as grid nodes. The Dirichlet boundary conditions are specified. Tab.2 shows the results of the numerical accuracy test and the ghost cells are defined by constant, linear, quadratic and cubic extrapolations. Fig.5 shows the numerical solution with 64 grid points and the ghost cells are defined by quadratic extrapolation.

4.2 Example 2

Consider $T_{xx} = f$ on $\Omega = [0, 0.5]$ with an exact solution of $T = 4x^2 \sin(2\pi x)$. The computational domain is discretized into cell of size Δx where the cell centers are referred to as grid nodes. The boundary conditions are given as $T(0) = \alpha$ and $T_x(0.5) = \beta$. Tab.3 shows the results of the numerical accuracy test and the ghost cells are defined by linear, quadratic and cubic extrapolations. Fig.6 shows the numerical solution with 64 grid points and the ghost cells are defined by cubic extrapolation.

Constant extrapolation

Number of points	L^∞- error	Order
16	0.13531	
32	0.06839	0.984
64	3.4395E-02	0.992
128	1.7200E-02	1.000
256	8.6382E-03	0.994

Linear extrapolation

Number of points	L^∞- error	Order
16	2.9232E-03	
32	7.3162E-04	1.998
64	1.8300E-04	1.999
128	4.5764E-05	2.000
256	1.1443E-05	2.000

Quadratic extrapolation

Number of points	L^∞- error	Order
16	1.1525E-05	
32	1.5525E-06	2.892
64	2.0306E-07	2.935
128	2.5958E-08	2.968
256	3.2811E-09	2.984

Cubic extrapolation

Number of points	L^∞- error	Order
16	3.2479E-07	
32	2.0668E-08	3.974
64	1.2974E-09	3.994
128	8.1219E-11	3.998
256	5.2820E-12	3.943

Table 2: The results of the numerical accuracy test and the ghost cells are defined by constant, linear, quadratic and cuic extrapolations.

Linear extrapolation

Number of points	L^{∞}- error	Order
16	8.2213E-04	
32	3.4943E-04	1.234
64	1.0548E-04	1.728
128	2.8641E-05	1.881
256	7.4445E-06	1.944

Quadratic extrapolation

Number of points	L^{∞}- error	Order
16	1.9985E-03	
32	5.0238E-04	1.992
64	1.2491E-04	2.008
128	3.1088E-05	2.006
256	7.7515E-06	2.004

Cubic extrapolation

Number of points	L^{∞}- error	Order
16	1.1580E-03	
32	1.4326E-04	3.015
64	1.7887E-05	3.002
128	2.2373E-06	2.999
256	2.7984E-07	2.999

Table 3: The results of the numerical accuracy test and the ghost cells are defined by linear, quadratic and cubic extrapolations.

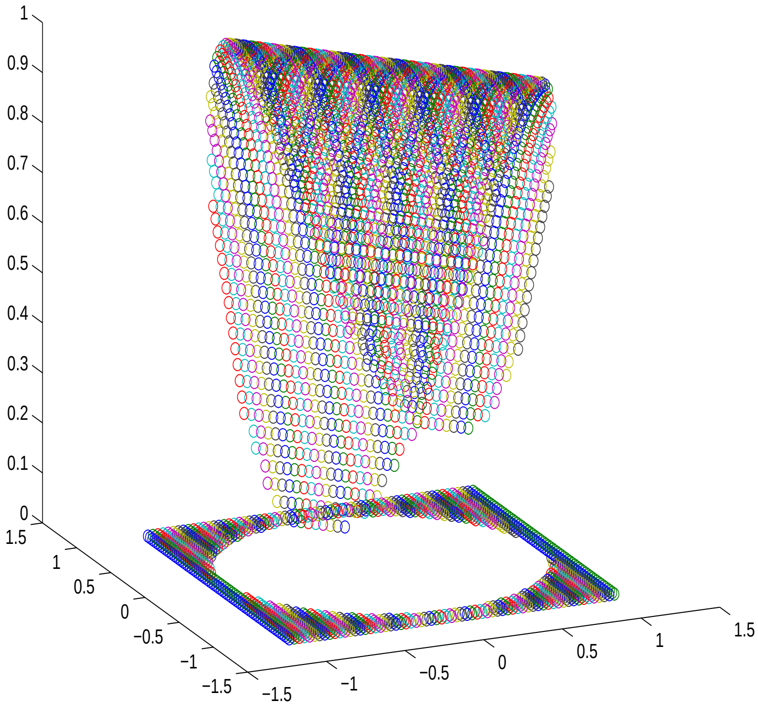

Figure 7: Solution of the Poisson equation on the unit circle. The exact solution is $T = \cos(x + y)$. The grid size is 64×64 and the ghost cells are defined by quadratic extrapolation.

4.3 Example 3

Consider the Poisson equation $\Delta T = -2\cos(x + y)$ on the unit circle with Dirichlet boundary conditions. The exact solution is $T = cos(x + y)$. The domain is embedded in a square. Outside the unit circle we set $T = 0$. Tab.4 shows the results of the numerical accuracy test and the ghost cells are defined by constant, linear, and quadratic extrapolations. Fig.7 depicts the solution on a 64×64 grid and the ghost cells are defined by quadratic extrapolation.

4.4 Example 4

Consider the Poisson equation $\Delta T = -\pi^2(\sin(\pi x)+\sin(\pi y)+\cos(\pi x)+\cos(\pi y)+30x^4 + 30y^4)$ on an irregular domain Ω^- with Dirichlet boundary conditions. The exact solution is $T = \sin(\pi x) + \sin(\pi y) + \cos(\pi x) + \cos(\pi y) + x^6 + y^6$. The domain is embedded in a square Ω. So we can regard the boundary of Ω^- as a interface that divies Ω into two disjoint pieces, Ω^- and Ω^+. Outside the interface we set $T = 0$.

Constant extrapolation

Number of points	L^∞- error	Order
16	6.6055E-02	
32	4.4915E-02	0.556
64	2.3703E-02	0.922
128	1.2745E-02	0.895

Linear extrapolation

Number of points	L^∞- error	Order
16	1.5867E-03	
32	3.3722E-04	2.234
64	9.6290E-05	1.808
128	2.2966E-05	2.068

Quadratic extrapolation

Number of points	L^∞- error	Order
16	3.2722E-04	
32	2.9732E-05	3.46017
64	4.2617E-06	2.80252
128	4.8286E-07	3.14175

Table 4: The results of the numerical accuracy test and the ghost cells are defined by constant, linear, and quadratic extrapolations.

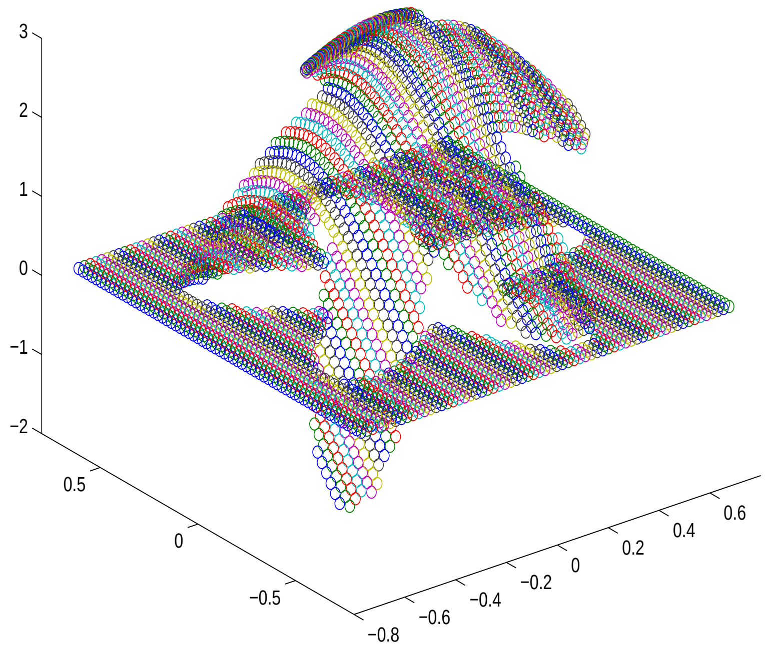

Figure 8: Solution of the Poisson equation on an irregular domain in two spatial dimensions. The exact solution is $T = \sin(\pi x)+\sin(\pi y)+\cos(\pi x)+\cos(\pi y)+x^6+y^6$. The grid size is 64×64 and the ghost cells are defined by quadratic extrapolation.

The interface is parameterized by $(x(\alpha), y(\alpha))$,where

$$\begin{cases} x(\alpha) = 0.02\sqrt{5} + (0.5 + 0.2\sin(5\alpha))\cos(\alpha), \\ y(\alpha) = 0.02\sqrt{5} + (0.5 + 0.2\sin(5\alpha))\sin(\alpha), \end{cases} \tag{4.18}$$

with $\alpha \in [0, 2\pi]$. Tab.5 shows the results of the numerical accuracy test and the ghost cells are defined by constant, linear, and quadratic extrapolations. Fig.8 depicts the solution on a 64×64 grid and the ghost cells are defined by quadratic extrapolation. Note that on irregular domains, the number of available grid nodes within the domain might limit the extrapolation to a lower degree for some grid resolution.

It is difficult to solve the 2D Poisson equation with Neumann boundary condition on irregular domain. Because we have no idea of the values of T_{xx} and T_{yy} on the boundary. So we can't extend the methodology discussed in Section 2.2 to two spatial dimensions.

Constant extrapolation

Number of points	L^∞- error	Order
16	0.2176	
32	0.1306	0.737
64	6.5600E-02	0.993
128	3.4000E-02	0.948

Linear extrapolation

Number of points	L^∞- error	Order
16	7.2809E-02	
32	3.4237E-03	4.410
64	8.2815E-04	2.048
128	2.0285E-04	2.030

Quadratic extrapolation

Number of points	L^∞- error	Order
16	6.4135E-02	
32	3.4870E-04	7.523
64	5.1107E-05	2.770
128	6.1969E-06	3.044

Table 5: The results of the numerical accuracy test and the ghost cells are defined by constant, linear, and quadratic extrapolations.

5 Conclusions

We have proposed a simple finite difference algorithm for obtaining higher-order accurate solutions for the Poisson equation subject to Dirichlet boundary conditions on irregular domains. The crucial issue is the discretization of the boundaries of the irregular domain. At the irregular point, we define ghost value constructed by extrapolation. In 1D Poisson equation with Dirichlet boundary condition problem, we get first, second, third and fourth order accuracy in the case of the constant, linear, quadratic and cubic extrapolations, respectively. In 1D Poisson equation with Neumann boundary condition problem, we get first, second and third order accuracy in the case of the linear and quadratic, and cubic extrapolations, respectively. In 2D Poisson equation with Dirichlet boundary condition problem, we get first, second and third order accuracy in the case of the constant, linear and quadratic extrapolations, respectively. And except the quadratic and cubic extrapolations, the linear system is symmetric.

References

[1] F. Gibou, R.P. Fedkiw, L.T Cheng, M Kang, A second-order accurate symmetric discretization of the Poisson equation on irregular domains. J. Comput. Phys. 176 (2002) 205–227.

[2] Frédéric Gibou, Ranald Fedkiw. A fourth order accurate discretization for the Laplace and Heat equations on arbitrary domains, with applications to the Stenfan problem. J. Comput. Phys. 202 (2005) 577–601.

[3] G. Golub and C. Van Loan, Matrix Computations. Johns Hopkins Univ. Press, Baltimore, 1989.

[4] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. J. Comput. Phys. 147 (1998) 60–85.

[5] H. Johansen, P. Colella, Cartesian grid embedded boundary finite difference methods for elliptic and parabolic differential equations on irregular domains. Ph. D. Thesis, University of California, Berkeley, CA, 1997.

[6] Petter Andreas Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions. J. Comput. Phys. 197 (2004) 364–386.

[7] R. Fedkiw, T. Aslam, B Merriman, S. Osher ,A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). J. Comput. Phys. 152 (1999) 457–492.

[8] R. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal 31 (1994) 1019–1044.

[9] Xu-Dong Liu, Ronald P. Fedkiw, and Myungjoo Kang, A boundary condition capturing mtheod for Poisson's equation on irregular domains. J. Comput. Phys. 160 (2000) 151–178.

[10] Y. Saad, Iterative Methods for Spares Linear System, PWS. Publishing Company, Boston, 1996.

[11] Z. Jomaa, C. Macaskill, The embedded finite difference method for the Poisson equation in a domain with an irregular boundary and Dirichlet boundary conditions. J. Comput. Phys. 202 (2005) 488–506.